数理解析研究所講究録
第 1666 巻 2009 年 49-56

行列のスペクトル分解アルゴリズムについて

田島 慎一
SHINICHI TAJIMA
新潟大学 工学部
FACULTY OF ENGINEERING, NIIGATA UNIVERSITY

飯塚 由貴恵
YUKIE IIZUKA
新潟大学 工学部
FACULTY OF ENGINEERING, NIIGATA UNIVERSITY

1 はじめに

行列のスペクトル分解は重要な概念であるが, 計算機代数の分野ではそれを正確に求めるアルゴリズムの研究は十分になされていないため, 行列のスペクトル分解アルゴリズムは既存の数式処理システムには実装されていない. 理論的には, レゾルベントの留数を用いることでスペクトル分解を表現する式を導けることが知られている. しかし, そのことをそのまま利用したプログラムでは逆行列の計算に時間がかかるため数式処理のプログラムとして実用性に乏しい.

本研究では, 最小多項式を基にスペクトル分解を計算する方法について考察した. 行列多項式の計算を行うことで, 逆行列の計算を避けスペクトル分解の固有値の多項式表現を求めるアルゴリズムを導出し, さらに数式処理システム Risa/Asir に実装した.

2 スペクトル分解

2.1 概念

A を n 次正方行列とし, 行列 A の最小多項式を $f(\lambda)$ で表す. 以下 $f(\lambda)$ が重複因子を持たない場合を考える. $f(\lambda)=0$ の相異なる解を α_j $(j=1,2,\ldots,l$ ただし $l \leq n)$ とおくと, 各 α_j に対して

$$\begin{cases} P_1+P_2+\cdots+P_l=E \\ \alpha_1 P_1+\alpha_2 P_2+\cdots+\alpha_l P_l=A \end{cases} \tag{1}$$

を満たす n 次正方行列 P_j が存在する. ここで, E は n 次単位行列である. P_j は射影行列と呼ばれ, 以下のようにレゾルベント $(\lambda E-A)^{-1}$ の極 α_j における留数として表現できる.

$$P_j=\frac{1}{2\pi i}\oint_{\alpha_j}(\lambda E-A)^{-1}d\lambda \tag{2}$$

ここで $(\lambda E-A)^{-1}$ をレゾルベントという. 式 (1) が行列 A のスペクトル分解である.

2.2 計算方法

2.2.1 準備

最小多項式 $f(\lambda)$ に対して

$$f(x) - f(y) = q(x,y)(x-y) \tag{3}$$

を考える. 式 (3) に $x = A,\ y = \lambda E$ を代入すると

$$f(A) - f(\lambda E) = q(A, \lambda E)(A - \lambda E) \tag{4}$$

を得る. ここで $f(A) = 0,\ f(\lambda E) = f(\lambda)E$ より

$$f(\lambda)E = q(A, \lambda E)(\lambda E - A) \tag{5}$$

となる. 式 (5) の両辺に右から $(\lambda E - A)^{-1}$ を掛けると

$$f(\lambda)(\lambda E - A)^{-1} = q(A, \lambda E) \tag{6}$$

であり, さらに, 式 (6) の両辺を $f(\lambda)$ で割ると

$$(\lambda E - A)^{-1} = \frac{1}{f(\lambda)} q(A, \lambda E) \tag{7}$$

となるので, 式 (2) に式 (7) を代入することで

$$P_j = \frac{1}{2\pi i} \oint_{\alpha_j} \frac{1}{f(\lambda)} q(A, \lambda E) d\lambda \tag{8}$$

を得る ([1, 4]).

2.2.2 $q(A, \lambda E)$ の計算

最小多項式 $f(\lambda)$ を

$$f(\lambda) = \lambda^l + a_{l-1}\lambda^{l-1} + \cdots + a_1\lambda + a_0 \tag{9}$$

で表されるモニックな l 次多項式とすると, $q(x, y)$ は

$$\begin{aligned}
q(x,y) &= \frac{f(x) - f(y)}{x - y} \\
&= \frac{(x^l + a_{l-1}x^{l-1} + \cdots + a_1 x + a_0) - (y^l + a_{l-1}y^{l-1} + \cdots + a_1 y + a_0)}{x - y} \\
&= \frac{(x^l - y^l) + a_{l-1}(x^{l-1} - y^{l-1}) + \cdots + a_2(x^2 - y^2) + a_1(x - y)}{x - y} \\
&= (x^{l-1} + x^{l-2}y + \cdots + xy^{l-2} + y^{l-1}) \\
&\quad + a_{l-1}(x^{l-2} + x^{l-3}y + \cdots + xy^{l-3} + y^{l-2}) + a_2(x + y) + a_1 \\
&= y^{l-1} + (x + a_{l-1})y^{l-2} + (x^2 + a_{l-1}x + a_{l-2})y^{l-3} \\
&\quad + \cdots + (x^{l-2} + a_{l-1}x^{l-3} + \cdots + a_2)y + (x^{l-1} + a_{l-1}x^{l-2} + \cdots + a_2 x + a_1)
\end{aligned} \tag{10}$$

となる. 式 (10) に $x = A,\ y = \lambda E$ を代入すると, $q(A, \lambda E)$ は

$$\begin{aligned}
q(A, \lambda E) &= E\lambda^{l-1} + (A + a_{l-1}E)\lambda^{l-2} + (A^2 + a_{l-1}A + a_{l-2}E)\lambda^{l-3} \\
&\quad + \cdots + (A^{l-2} + a_{l-1}A^{l-3} + \cdots + a_2 E)\lambda + (A^{l-1} + a_{l-1}A^{l-2} + \cdots + a_2 A + a_1 E)
\end{aligned} \tag{11}$$

となる. さらに, 式 (11) の変数 λ の次数ごとの係数行列に注目すると

$$\begin{aligned} q(A,\lambda E) &= E\lambda^{l-1} + (A + a_{l-1}E)\lambda^{l-2} + ((A + a_{l-1}E)A + a_{l-2}E)\lambda^{l-3} \\ &\quad + \cdots + (\cdots(((A + a_{l-1}E)A + a_{l-2}E)A + a_{l-3}E)A + \cdots + a_2E)\lambda \\ &\quad + ((\cdots(((A + a_{l-1}E)A + a_{l-2}E)A + a_{l-3}E)A + \cdots + a_2E)A + a_1E) \end{aligned} \tag{12}$$

と変形でき, k 次の係数行列は $k+1$ 次の計算結果を用いて表現できることがわかる. 従って, ホーナー法を用いることで係数行列を効率的に計算できる ([2, 6]).

2.2.3 導出

$f(\lambda)$ は重複因子を持たないので, $f(\lambda)$ と $f'(\lambda)$ は互いに素であることから

$$a(\lambda)f(\lambda) + b(\lambda)f'(\lambda) = 1 \tag{13}$$

を満たす多項式 $a(\lambda)$, $b(\lambda)$ が存在する. よって, 式 (8) は

$$\begin{aligned} P_j &= \frac{1}{2\pi i}\oint_{\alpha_j} \frac{a(\lambda)f(\lambda) + b(\lambda)f'(\lambda)}{f(\lambda)} q(A,\lambda E)d\lambda \\ &= \frac{1}{2\pi i}\oint_{\alpha_j} \left(a(\lambda)q(A,\lambda E) + \frac{f'(\lambda)}{f(\lambda)} b(\lambda)q(A,\lambda E) \right) d\lambda \end{aligned} \tag{14}$$

と書くことができる.

次に, 式 (14) の被積分関数中の特異性に注目する. 第一項は $\lambda = \alpha_j$ で正則であり, 第二項は $\lambda = \alpha_j$ に極を持つことから

$$P_j \equiv \frac{1}{2\pi i}\oint_{\alpha_j} \left(\frac{f'(\lambda)}{f(\lambda)} b(\lambda)q(A,\lambda E) \right) d\lambda \tag{15}$$

を得る. さらに, 留数計算の公式

$$\frac{1}{2\pi i}\oint_{\alpha} \frac{f'(x)}{f(x)} h(x)dx = h(\alpha) \tag{16}$$

を用いると, 射影行列 P_j は

$$P_j = b(\alpha_j)q(A,\alpha_j E) \tag{17}$$

で与えられることがわかる (cf. [3, 7]). このように, 各々の固有値 α_j に対して射影行列 P_j は式 (17) のように共通の表現となることから, 固有値 α_j を不定文字 λ に置き換えることにする. ここで $f(\alpha_j) = 0$ より, 式 (17) の右辺の α_j を λ に置き換えたものに対して $f(\lambda)$ で割った余りを考え, これを $P(A,\lambda)$ とすると

$$P(A,\lambda) = b(\lambda)q(A,\lambda E) \ mod \ f(\lambda) \tag{18}$$

となる. l 次最小多項式 $f(\lambda)$ に対して $b(\lambda)$ と $q(A,\lambda E)$ はともに $l-1$ 次多項式であることから $b(\lambda)q(A,\lambda E)$ は $2l-2$ 次多項式となるが, $f(\lambda)$ で割った余りを考えることで $P(A,\lambda)$ を λ の高々 $l-1$ 次多項式で表現できる.

3 作成したプログラム

3.1 プログラムの仕様

- 数式処理システム Risa/Asir ([5]) で動作するプログラム

- 機能：最小多項式が重複因子を持たない場合の行列のスペクトル分解を計算
- 呼出し形式：関数名 (Mat, Poly)
- 引数：Mat ← 正方行列 , Poly ← 特性多項式
- 戻り値：[$b(\lambda)$ の共通分母 , [λ^{l-1} の係数行列 $\cdots$ λ の係数行列 定数項の係数行列]]

$P(A,\lambda)$ が戻り値であり,

$$P(A,\lambda)=\frac{1}{b(\lambda)\text{ の共通分母}}((\lambda^{l-1}\text{の係数行列})\lambda^{l-1}+\cdots+(\lambda\text{の係数行列})\lambda+(\text{定数項の係数行列})) \quad (19)$$

のように戻り値の要素と対応している. 従って, 式 (19) の不定文字 λ を固有値 α_j に置き換えることで, 固有値 α_j が定める固有ベクトル空間への射影行列 P_j が exact にわかる.

3.2 具体例

$$A=\begin{pmatrix}-4&5&-4\\-3&3&3\\-1&-2&3\end{pmatrix},\ E=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},\ f(\lambda)=\lambda^3-2\lambda^2+2\lambda+66 \quad (20)$$

に対してスペクトル分解を計算する. このとき

$$f'(\lambda)=3\lambda^2-4\lambda+2 \quad (21)$$

$$b(\lambda)=\frac{1}{60134}(2\lambda^2-303\lambda+202) \quad (22)$$

$$q(A,\lambda E)=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\lambda^2+\begin{pmatrix}-6&5&-4\\-3&1&3\\-1&-2&1\end{pmatrix}\lambda+\begin{pmatrix}15&-7&27\\6&-16&24\\9&-13&3\end{pmatrix} \quad (23)$$

であり, $P(A,\lambda)$ は

$$\begin{aligned}&P(A,\lambda)\\&=\frac{1}{60134}\left(\begin{pmatrix}1424&-1509&1250\\909&-731&-849\\317&572&-693\end{pmatrix}\lambda^2+\begin{pmatrix}-5267&3111&-8973\\-2412&5512&-6678\\-2925&3543&-245\end{pmatrix}\lambda+\begin{pmatrix}23556&-2074&5982\\1608&16370&4452\\1950&-2362&20208\end{pmatrix}\right)\end{aligned} \quad (24)$$

となる. 式 (24) に固有値 $\lambda=\alpha_1,\alpha_2,\alpha_3$ をそれぞれ代入すると射影行列 P_1,P_2,P_3 がわかり, スペクトル分解が求まる.

3.3 入出力例

3.2 節の行列に対する入出力例を示す. あらかじめ関数 spect (仮の関数名) をロードする必要がある.

```
[0] [14] A=newmat(3,3,[[-4,5,-4],[-3,3,3],[-1,-2,3]]);
[ -4 5 -4 ]
[ -3 3 3 ]
[ -1 -2 3 ]
[15] E=newmat(3,3,[[1,0,0],[0,1,0],[0,0,1]]);
[ 1 0 0 ]
[ 0 1 0 ]
[ 0 0 1 ]
[16] F=det(x*E-A);
x^3-2*x^2+2*x+66
[17] spect(A,F);
[60134,[ [ 1424 -1509 1250 ]
[ 909 -731 -849 ]
[ 317 572 -693 ] [ -5267 3111 -8973 ]
[ -2412 5512 -6678 ]
[ -2925 3543 -245 ] [ 23556 -2074 5982 ]
[ 1608 16370 4452 ]
[ 1950 -2362 20208 ] ]]
[18]
```

ここで, 関数 `spect` の出力結果内の 60134 は, 3.2 節の $b(\lambda)$ の分母であり, $P(A,\lambda)$ の分母と一致する. また, 特性多項式を求める際に行列のサイズが大きい場合は, 京都大学の木村欣司氏が作成した特性多項式計算プログラムを用いる.

4 アルゴリズム

行列のスペクトル分解を求めるアルゴリズムの流れは以下の通りである.

1. 最小多項式 $f(\lambda)$ が重複因子を持たないことを確かめる. 重複因子を持つ場合は, エラーメッセージを表示し終了させる.
2. $b(\lambda)$ を求める.
3. $q(A,\lambda E)$ を求める.
4. $P(A,\lambda)=b(\lambda)q(A,\lambda E)\ mod\ f(\lambda)$ を求め, $P(A,\lambda)$ を戻り値として出力する.

ここで, $q(A,\lambda E)$ の変数 λ と $b(\lambda)$ に対して

$$\lambda^{k+1}b(\lambda)\ mod\ f(\lambda)=\lambda(\lambda^k b(\lambda)\ mod\ f(\lambda))\ mod\ f(\lambda) \tag{25}$$

より, 次数 k における計算結果を用いることで, 次数 $k+1$ における計算を効率的に行うことができる. その他, 計算速度を向上させるために以下の工夫を行った.

- 最小多項式 $f(\lambda)$ の変数を扱わずに, 各次数毎の係数行列のみを計算する.
- 多項式 $b(\lambda)$ の係数が有理数になることに注目し, $b(\lambda)$ を係数の共通分母となる整数と, 整数係数多項式に分けた ($b(\lambda)$ は [整数係数多項式, 共通分母] の形で扱われている).

- $q(A, \lambda E)$ の係数行列の計算等, 行列多項式計算を行う場合はホーナー法を利用する (2.2.2 節参照).
- $\lambda^k b(\lambda)\ mod\ f(\lambda)$ を先に計算し, その結果に $q(A, \lambda E)$ の係数行列を掛けることで, 乗算の回数を少なくした.

5 計算時間の測定・考察

行列のスペクトル分解を求めるプログラムの CPU 時間, GC 時間, 経過時間を測定した.

5.1 測定環境

- 使用ソフト … 数式処理システム Risa/Asir
- 実験で使用したコンピュータ
  - OS … Windows XP
  - CPU … Intel(R) Pentium(R) 4 (2.52GHz)
  - メモリ … 2.00GB

5.2 測定方法

- 行列の要素は整数であり, -512〜511 (3 桁以内), -65536〜65535 (5 桁以内), -8589934592〜8589934591 (10 桁以内) の 3 パターンで実験
- n 次正方行列に対して 10 回ずつ測定し, それぞれの平均をとる
- CPU 時間, GC 時間, 経過時間は Asir の time() で取得
- Asir の設定はデフォルトのまま

5.3 測定結果

表 1 : スペクトル分解の計算にかかる時間 (行列の要素 : 3 桁以内)

n	CPU 時間 (sec)	GC 時間 (sec)	経過時間 (sec)	$\frac{CPU\text{ 時間}}{\text{経過時間}}$ (%)
10	0.0359375	0.0140625	0.0501000	71.7
15	0.192188	0.0953125	0.293800	65.4
20	1.07188	0.420313	1.50630	71.2
25	4.23282	1.16250	5.42650	78.0
30	13.8641	2.45469	16.4107	84.5
35	38.6094	7.19375	46.2452	83.5
40	92.1938	13.9922	106.591	86.5
45	213.795	26.9969	241.917	88.4
50	430.066	41.3109	474.849	90.6
53	636.452	56.4594	696.366	91.4

表 2 : スペクトル分解の計算にかかる時間 (行列の要素 : 5 桁以内)

n	CPU 時間 (sec)	GC 時間 (sec)	経過時間 (sec)	$\frac{CPU\text{ 時間}}{\text{経過時間}}$ (%)
10	0.0500000	0.0390625	0.0936000	53.4
15	0.415625	0.231250	0.653200	63.6
20	2.22969	0.710938	2.96080	75.3
25	9.29063	1.57813	10.9218	85.1
30	31.2672	4.34219	35.7625	87.4
35	92.1391	10.2188	102.756	89.7
40	224.383	17.1453	242.478	92.5
43	362.433	23.7469	388.195	93.4

表 3 : スペクトル分解の計算にかかる時間 (行列の要素 : 10 桁以内)

n	CPU 時間 (sec)	GC 時間 (sec)	経過時間 (sec)	$\frac{CPU\text{ 時間}}{\text{経過時間}}$ (%)
10	0.115625	0.0937500	0.209400	55.2
15	1.14063	0.501563	1.66090	68.7
20	7.22188	2.06875	9.33430	77.4
25	31.8344	4.18594	36.1719	88.0
30	106.700	7.08125	114.225	93.4
35	315.308	16.1406	333.127	94.7
40	784.474	25.8406	816.046	96.1
42	1091.44	30.8250	1128.48	96.7

行列の要素が 3 桁以内のときは $n = 53$, 5 桁以内のときは $n = 43$, 10 桁以内のときは $n = 42$ を超えると "Too many heap sections" と表示され, 計算ができなかった.

5.4 考察

行列の要素の桁数を変えて実験を行ったところ, $n = 40$ において要素が 10 桁以内のときの経過時間は, 5 桁以内のときと比較して約 3.4 倍かかった. 時間こそかかるものの, 計算可能な行列のサイズは 5 桁以内のときとそれほど差が無い.

次に経過時間に占める CPU 時間の割合に注目する. 3 つの表を比べると, 行列のサイズが小さいときは行列の要素の桁数が少ないときの方が割合が高いが, サイズが大きくなると桁数の多い場合に CPU 時間の割合が高くなる傾向がある.

6 まとめ

逆行列の計算を避けることで, スペクトル分解を求めることが可能になった. また, 今回作成したプログラムは固有値の多項式として射影行列を表現する式を求めており, 固有値の値を代入することでスペクトル分解を正確に求めることが可能となっている.

2009 年 1 月, 金沢大学の小原功任氏との共同研究により計算効率の良い新たなアルゴリズムを導出し, Risa/Asir に実装した. また 2009 年 3 月には, 最小多項式が因数分解できる場合 (重複因子を持たない), そのことを利用してスペクトル分解を求めるアルゴリズムを実装した. 以上 2 つのアルゴリズムについては別の機会に詳しく述べることにする.

7 今後の課題

本アルゴリズムは最小多項式が重複因子を持たない場合に限定していたので, 今後最小多項式が重複因子を持つ場合のスペクトル分解の計算法について研究をする予定である. この場合はレゾルベントが 2 位以上の極を持つため本アルゴリズムをそのままの形で適用することはできない. したがって, 本アルゴリズムを拡張していく必要がある.

本研究は, 平成 20 年度 内田エネルギー科学振興財団の研究助成を受けている.

参 考 文 献

[1] F. シャトラン (伊理 正夫・伊理 由美 訳) : 行列の固有値 新装版 最新の解法と応用, シュプリンガー・フェアラーク東京 (2003).

[2] F. R. Gantmacher : The Theory of Matrices, vol.1, 2, Chelsea (1960).

[3] 加藤 涼香, 田島 慎一 : 有理関数のローラン展開アルゴリズムと代数的局所コホモロジー, 京都大学数理解析研究所講究録 1395 「Computer Algebra - Design of Algorithms, Implementations and Applications」(2004), pp.50-56.

[4] T. Kato : A Short Introduction to Perturbation Theory for Linear Operators, Springer-Verlag (1982).

[5] M. Noro and T. Takeshima, Risa/Asir - a computer algebra system, ISSAC 1992 (ed. P. S. Wang), ACM (1992), pp.387-396.

[6] 庄司 卓夢, 田島 慎一 : 多項式剰余公式の計算アルゴリズム, 京都大学数理解析研究所講究録 1514 「Computer Algebra - Design of Algorithms, Implementations and Applications」(2006), pp.108-114.

[7] 田島 慎一 : 多変数補間問題とホロノミック D-加群, 千葉大学数学セミナーノート No.3 「代数解析学の諸問題」(1999), pp.73-94.